Machine Learning

(Course 41204)

Winter 2023

Professor Malden Kolar

Stroke Prevention Exercise for the Department of Health and Human Services

Diemeng Hu

Chloe Zhang

Griffin Ong

Julia Wood

I. Introduction

Motivation

Stroke is the 5th leading cause of death and a leading cause of serious, long-term disability, with an estimated cost of $34 billion annually.[1] According to the Centers for Disease Control and Prevention, 800,000 people have a stroke every year, more than 140,000 die, and many survivors face disability. However, more than 80% of strokes are preventable through the identification of risk factors, preventative treatment, and healthy lifestyle changes.[1]

Patients who undergo stroke rehabilitation and recovery services incur a substantial financial loss. According to the American Heart Association, Americans spend an annual average of $45.5 billion on direct and indirect costs of stroke. Within this amount, Americans spend $7.9 billion on hospital inpatient stays for stroke, $2.4 billion on hospital outpatient or office-based provider visits for stroke, and $8.2 billion for home health care for stroke. In addition, stroke costs Americans $17.5 billion each year in lost wages.[2]

Per request of the Department of Health and Human Services, we have designed a machine learning model that predicts the probability of whether someone will have a stroke based on relevant input parameters. Our hope is the Department of Health and Human Services will share this model with US-based government funded hospitals so healthcare practitioners can identify which patients are at high-risk for stroke and prescribe effective treatments like Atorvastatin, which reduce the risk of stroke by lowering the amount of cholesterol and other fats in the blood.[3]

II. Data Visualization and Preprocessing

The raw stroke prediction dataset was obtained from Kaggle[4] and contains 5,110 patient observations with 11 variables per observation. These features have valuable information about the different patient characteristics (i.e., gender, age, marriage, working status, and residential location) and the lifestyle and case history each patient (e.g., BMI, smoking or not, glucose level, hypertension and heart disease). While the data are relatively clean, some variables have inconsistent formats and convoluted structures, all of which we address through data processing.

Before plotting summary statistics, we first identified missing values and converted variable types as appropriate. All observations with a missing BMI value were removed and converted to numeric values. In order to conform to the generality of the model, character type variables, including gender (male/female), hypertension (0 = no hypertension, 1 = hypertension), heart disease (0 = no heart disease, 1 = heart disease), working status (never worked, children, self-employed, government job, private job), ever married (yes/no), and smoking status (formerly smoked, never smoked, smokes, unknown), and the dependent variable stroke (yes/no) were converted to factor variables.

Visualization

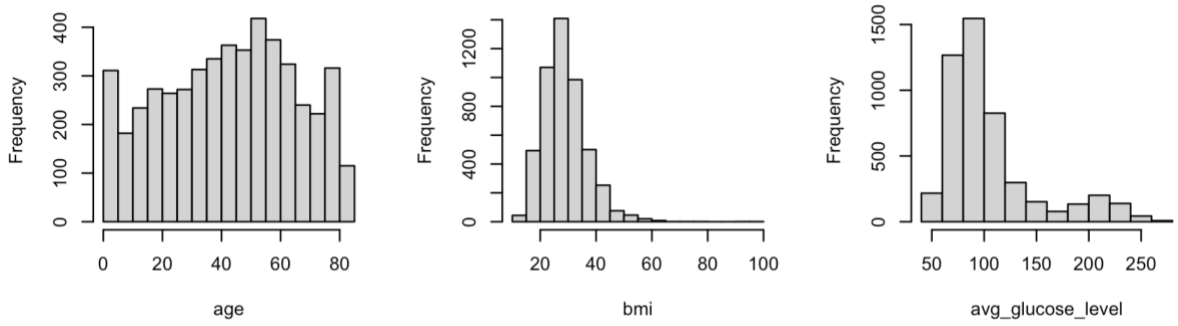

Figure 1. Continuous Variable Frequency Distributions

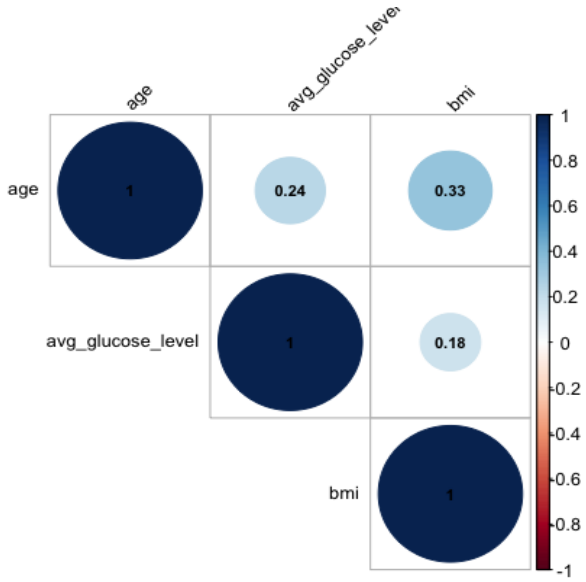

Figure 2. Continuous Variable Correlation Matrix

In the dataset, only three variables are continuous. Among these three, age appears to follow a relatively normal distribution, whereas the variables *BMI* and *avg_glucose_level* are negatively skewed as depicted in Figure 1. Initially, we contemplated transforming the data to create a range of more evenly distributed covariate values. However, as our model demonstrated accuracy on the test dataset, such transformation was deemed unnecessary. Figure 2 shows the correlation coefficients between the numerical variables is low, so all the continuous variables should be considered into the model.

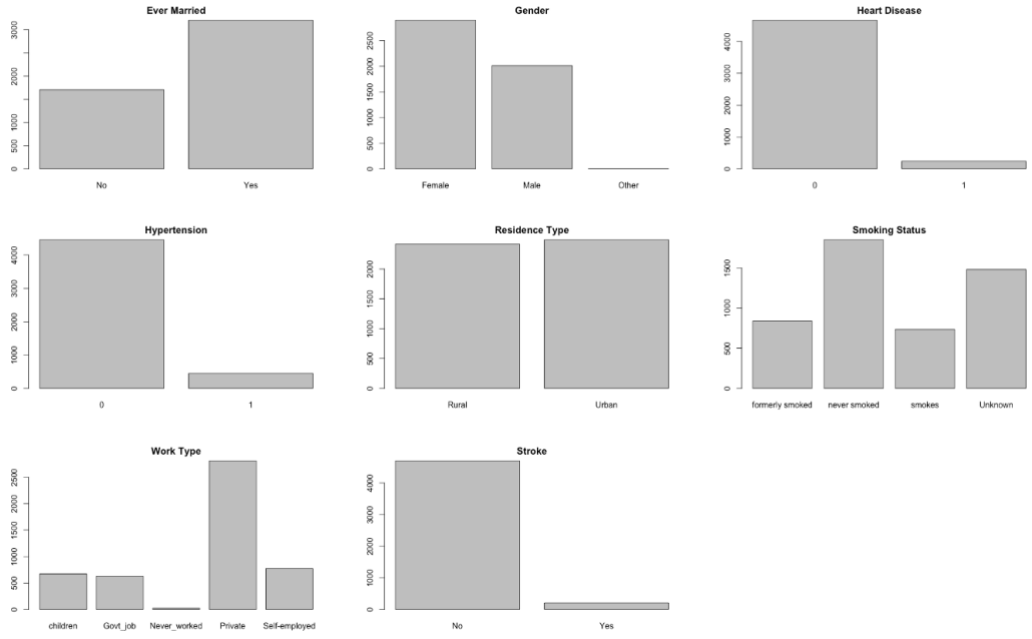

Figure 3. Barplots of Categorical Variables

As shown in Figure 3, the distribution of the dependent variable, *stroke*, exhibits a significant imbalance in the data, with a ratio of 22 to 1 between the number of individuals who do not experience a stroke ("no") and those who do ("yes"). To address this imbalance, we resample the data to include more instances of stroke occurrences. In Figure 3, we note that the variable *gender* has only one observation in the "other" category. Given the limited sample size, we removed this data point with the important caveat that the conclusions from our analysis are only applicable to the population of patients who identify as male/female. In addition, both *hypertension* and *heart disease* exhibit a small proportion of patients with a value of 1, indicating the presence of these conditions. Similarly, the *work_type* variable has only 22 patients who report never working. Notably, the smoking status of 1483 patients was unknown. This is an

ordinal categorical variable. As a result, it should be treated as a quantitative variable and may be inappropriate to assume linearity in our analysis. Further investigation is warranted to determine the most appropriate approach to model this variable.

Balancing

Before fitting any models, we first split the data and balanced the training set. Using the stratified sampling method, we created a training set (70% of the data) and a validation set (30% of the data). We trained our models with the training set and selected the best performing parameters. We used the validation set to fit our final models.

After splitting, we resampled and balanced the data using the ROSE function. After balancing, the ratio of patients that had a stroke versus patients that did not have a stroke in the data set is 1,664:1,773, which is reasonable.

III. Modeling

The project uses supervised learning algorithms to analyze and predict data. To achieve this, we intend to train datasets on a range of models, including SVM, random Forest, KNN, logistic regression, boost, and neural networks. Given that the choice of parameter values is critical in the development of every model, we will explore a range of values to select the best values for optimal model performance. To measure the performance of each model, we will use various metrics such as accuracy, misclassification error, AUC scores, ROC plot, and confusion matrix. We will analyze performance evaluation metrics to evaluate the applicability of the model. This evaluation will help us determine the most appropriate model.

Meanwhile, we will conduct a predictor ranking analysis to determine the relative importance of each predictor variable in our models. By using the <varImp> function available in the <caret> package of R, we will identify those with the greatest impact on model performance. This step is critical in identifying the key factors that contribute to stroke and will enable healthcare providers to focus on these factors when treating patients to prevent stroke.

SVM

We employed the "svmRadial" method to build an SVM model for predicting the occurrence of stroke. The model was trained using the repeated k-fold cross-validation technique with 10 folds and 3 repetitions. The model was then tuned by testing it with 10 different values of sigma and C, and the best performing parameters were chosen based on the highest accuracy. Overall, the SVM model showed promising results with an accuracy of 87.48% and a kappa of 0.7498 using the optimal parameter values of sigma = 0.028 and C = 64, where accuracy is the ratio of correctly classified instances to the total number of instances, while kappa is a measure of agreement between the predicted and actual outcomes, which considers the chance agreement.

Random Forest

Random Forest algorithm was employed to train the stroke classification model using the "rf" method. To train the model, we used the training dataset and a repeated cross-validation method with 10 folds and 3 repeats. This method helps to avoid overfitting and provides a more reliable estimate of the model's performance. We used the "tuneGrid" parameter to set the grid of hyperparameters, and the "mtry" parameter was varied over a range of values from 1 to 15. The results of the hyperparameter tuning show that the optimal value of "mtry" was 4, which resulted in an accuracy of 84.46% and a kappa of 0.69. The "bestTune" function output shows that the model's best performing value for "mtry" was 4, which provides valuable insights into the optimal hyperparameter setting.

KNN Model

We employed KNN modeling on the training dataset using the “knn” method. Similar to random forest, we trained the model using the K-fold cross-validation with 10 folds and 3 repeats to reduce variability in model performance and identify the best tuning parameters. Leveraging the “tuneLength” parameter, the model was tuned with 20 different values of k ranging from 5 to 43 in odd numbers. The results of hyperparameter tuning show the optimal value of k is 41 which provides an accuracy of 77.02%.

Logistic Regression

We employed standard logistic regression modeling on the training set using the “glm” function and K-fold cross validation with 10 folds and 3 repeats. We opted not to use coefficient

shrinkage methods (i.e., lasso, ridge) due to the limited quantity of variables within our data set. The logistic regression model showed an accuracy of 77.25% and a kappa value of 0.5455, the level of agreement between predicted and actual outcomes. Notice this value of kappa is less than the SVM model (.7498).

Boosted Tree

We employed boost modeling on the training dataset using the "xgboost" function. We created a grid that set a variety of values for the shrinkage rate (.1, .01, .001), interaction depth (1, 2, 4), and number of rounds (1000, 2000, 5000). Typically, the smaller the shrinkage rate the larger number of trees needed in order to achieve optimal performance. The results of the hyperparameter tuning are a shrinkage rate of 0.01, an interaction depth of 4, and 1000 rounds. The model accuracy is 75.4%.

Neural Network

We used the neural network method with the <nnet> package in R to predict stroke occurrence based on a set of ten predictor variables. We utilized a repeated cross-validation technique with ten folds and three repeats, and grid search for hyperparameter tuning. We specified a tuning grid with three values for the size of the hidden layer and three values for the weight decay parameter. Based on the accuracy metric, the optimal model was selected with a hidden layer size of 15 and weight decay of 0.1. The final neural network model had 271 weights and achieved an accuracy of 85.58% and a kappa coefficient of 0.711 on the test set.

IV. Results

Below, we present the misclassification rates, accuracy rates, AUCs, and ROC curves for our selected SVM, random forest, KNN model, logistic regression model, boosting, and neural network model on the 1,472 test set observations. A probability threshold of .5 was used to predict the results for all models.

Model	Misclassification Rate	Accuracy Rate	AUC
SVM	0.192	0.808	0.666
Random Forest	0.205	0.795	0.806
KNN	0.268	0.732	0.830
Logistic Regression	0.242	0.758	0.842
Boosting	0.246	0.754	0.834
Neural Network	0.192	0.808	0.674

Table 1. Misclassification Rates, accuracy rate and AUC of models

According to Table 1, the SVM and neural network appear to be the best-performing models on the test dataset with accuracy rates of 80%, followed by the random forest (79.5%) and boosting (75.4%). Surprisingly, however, when comparing AUCs, logistic regression and KNN perform the best (84.2% and 83% respectively), whereas SVM and neural network perform the worst (66.6% and 67.4%). Both the SVM model (highest accuracy) and KNN model (highest AUC) project similar variable importance (Figure 5 and Figure 6), with hypertension, average glucose level, age, and marital status being the most important predictors. In contrast, the logistic regression model projects hypertension, average glucose level, age, and heart disease as the most important predictors.

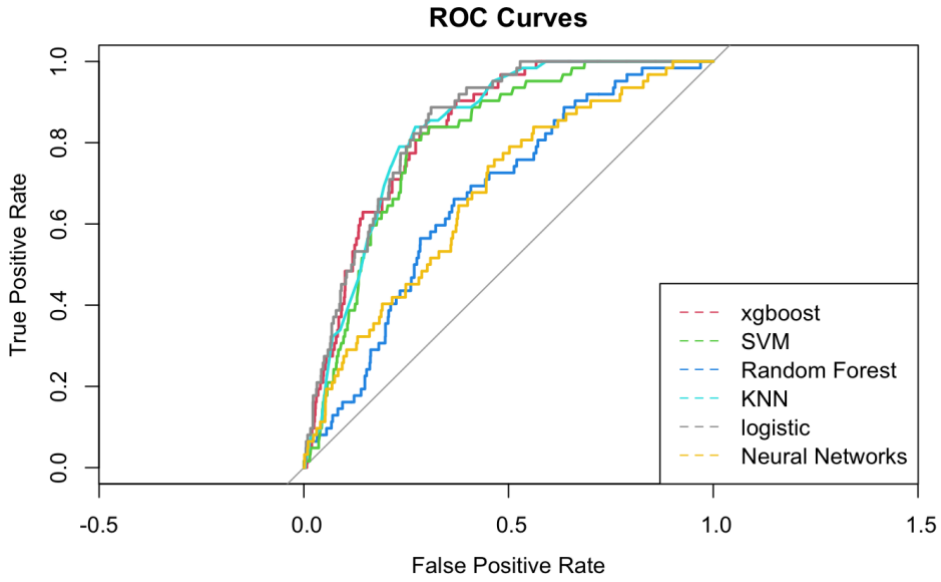

Figure 4. ROC Curves

The significant discrepancy in results between which models have a better accuracy rate vs. AUC score is due to the test data being extremely imbalanced, with a ratio of 23 negative observations to 1 positive observation. The accuracy rate tells us how well the model is at classifying the majority class, where AUCs measure how well the model performs across all thresholds. SVM and neural networks have a harder time distinguishing the minority class from the majority class, which is reflected in their corresponding AUCs and positioning on the ROC curve (Figure 4).

Given the importance of identifying the minority class (i.e., patients who are at risk) for the DHHS, we decided model selection should be based on the corresponding confusion matrices (Table 2) versus accuracy rates. This is discussed in more detail in our analysis below.

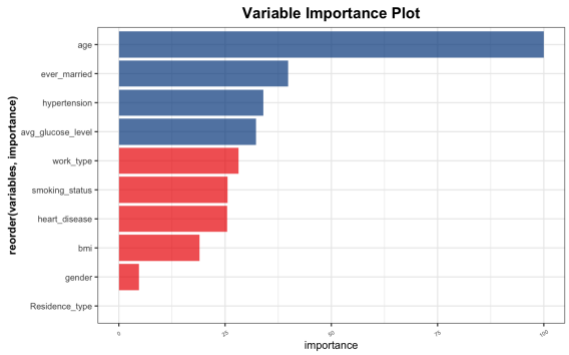

Figure 5. SVM Model Variable Importance Plot

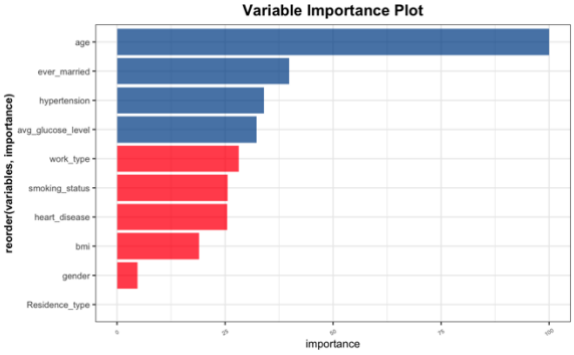

Figure 6. KNN Model Variable Importance Plot

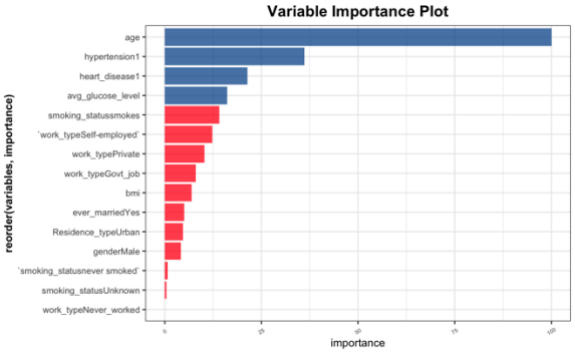

Figure 7. Logistic Regression Model Variable Importance Plot

SVM		
	Yes	No
Yes	1170	44
No	240	18

Random Forest		
	Yes	No
Yes	1132	23
No	278	39

KNN		
	Yes	No
Yes	1026	10
No	384	52

Logistic Regression		
	Yes	No
Yes	1068	14
No	342	48

Boosting		
	Yes	No
Yes	1065	17
No	345	45

ANN		
	Yes	No
Yes	1168	40
No	242	22

Table 2. Confusion Matrices

V. Analysis

Misclassification and accuracy rates reduce classifier performance to a single number, which is too simplistic for our purposes (i.e., false positive predictions may be preferred because of the high costs of not identifying a high-risk patient before they have a stroke). In addition, as mentioned previously our test data is extremely unbalanced. Given this, we assigned costs to both false negative and false positive results in the confusion matrices (Table 2) and evaluated the total costs of each model accordingly to select the best model for the DHHS.

For our purposes, a false negative is defined as not catching someone who is at high risk for a stroke after performing critical testing of a patient's hypertension and average glucose levels and failing to prescribe them Atorvastatin. Assuming they end up in the hospital with one stroke in a given year, the average cost of a hospital stays (i.e., the average cost of a false negative) is approximately $32,024.[2] In contrast, a false positive is treating a patient with Atorvastatin to prevent a stroke when they do not need it because they are not at risk. The cost of prescribing Atorvastatin for one year (i.e., the cost of a false positive) is approximately $1500.[6] Under these assumptions, the result of table 3 is calculated based on the follow formula:

$$Total\ Cost = \frac{\#\ of\ false\ negative}{\#\ of\ total\ patients} \times (-\$32{,}024) + \frac{\#\ of\ false\ positive}{\#\ of\ total\ patients} \times (-\$1{,}500).$$

Based on the costs associated with Type I and Type II errors of these models, it can be concluded that KNN is the best performing model. as it is the least costly at $608 (Table 3).

Model	False Negative Costs	False Positive Costs	Total Costs
SVM	-957.24	-244.57	-1201.80
Random Forest	-500.38	-283.29	-783.66
KNN	-217.55	-391.30	-608.86
Logistic Regression	-304.58	-348.51	-653.08
Boosting	-369.84	-351.56	-721.40
Neural Network	-870.22	-246.60	-1116.82

Table 3. Estimated costs produced by models

VI. Conclusion

SVM and neural networks were the most accurate models in stroke prediction when applied to the test dataset, and KNN is the best performing model with respect to minimizing expected costs. Regardless of model, we recommend each US-based government funded hospital recalculate expected probabilities, costs, and benefits of each model leveraging their own proprietary data. The hope is that the data is as balanced as possible to produce the most accurate insights into which model is the strongest. The model that they determine is the strongest should inform what variables they track in patients to determine whether they are at high risk for a stroke.

VII. Extensions & Potential for Additional Analysis

There are several opportunities for development that would allow us to predict stroke risk more comprehensively and accurately.

Retrain Model with Local Data and Larger Number of Observations: Given the data was made readily available via open source on Kaggle, hospitals that receive the model from the DHHS should retrain the model using their own proprietary data to ensure the validity of the model's performance within their patient population. Moreover, hospitals can incorporate other variables like race and prior health history to strengthen the model. Previous literature suggests

that factors like race, cholesterol, and sickle cell disease are correlated with stroke, but that information was not available within the data used for this analysis.[5] In addition, the initial data used for the analysis was severely unbalanced, so collecting more observations to train the model will further help to refine it. The Center for Medicare and Medicaid Services (CMS) or Veterans Affairs could be potential partners for DHSS to obtain data at a federal level. Looking down the line towards more local healthcare providers, these smaller health systems should opt to utilize state, county, or regional patient data that best reflects the communities they serve. In doing so, providers will have access to a certain degree of model specialization and flexibility that allows them to offer the best care that delivers on community needs.

Incorporate Model into Electronic Medical Records (EMR) for Live Stroke Predictions: EMRs are the prevailing method for storing patient health information. When physicians meet with a patient, they enter health data into the EMR, a "one-stop shop" system that allows hospital staff to view all the patient's health history at a glance. A model like ours can easily be applied to EMR systems across providers - the model can automatically utilize the data in the EMR to assess a patient's stroke risk instantaneously. As a result, there are no delays in identifying and discussing stroke risks with the patient. This may catch strokes well before they happen and initiate preventive care early. Further, with access to the entirety of the patient's medical file, the model can identify new important variables outside of our dataset that are strong predictors of stroke. This provides more observations for the model to calibrate itself, ultimately leading to better predictions down the line.

Incorporate "Smart" Health Technology: Similar to the EMR, technology within Apple, Samsung, or similar smart devices that monitor health data can be utilized to alert patients that they are at risk for stroke. Then, individuals can consult with their physician about charting a path forward for mitigating stroke risk. Here, as well as in the EMR case, the more individuals buy into the services and share their data, the more enhanced the models will become, and the more impact these models can have on patient health outcomes. This goes hand-in-hand with our initial recommendations for more data, all of which can be fed into complex and accurate models.

References

1. “Preventing Stroke Deaths | VitalSigns.” 2017. CDC. https://www.cdc.gov/vitalsigns/stroke/index.html.

2. “Covering the cost of stroke - Washington National Insurance Blog.” 2020. Washington National Insurance Company. https://washingtonnational.com/explore/why-insurance/how-to-cover-stroke-cost/.

3. Tracer, Howard. 2021. “Medicines to Prevent Heart Attack and Stroke: Questions for the Doctor - MyHealthfinder | health.gov.” Office of Disease Prevention and Health Promotion. https://health.gov/myhealthfinder/doctor-visits/talking-doctor/medicines-prevent-heart-attack-and-stroke-questions-doctor.

4. “Stroke Prediction Dataset.” n.d. Kaggle. Accessed March 8, 2023. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

5. “Know Your Risk for Stroke”. CDC. 2022. https://www.cdc.gov/stroke/risk_factors.htm.

6. Bawab, Josephine. n.d. “How much is atorvastatin without insurance?” SingleCare. Accessed March 9, 2023. https://www.singlecare.com/blog/atorvastatin-without-insurance/.